UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/086,032	02/28/2002	Satish Ramchandra Thatte	MS196129.1	5174

27195 7590 11/05/2004

AMIN & TUROCY, LLP
24TH FLOOR, NATIONAL CITY CENTER
1900 EAST NINTH STREET
CLEVELAND, OH 44114

EXAMINER	
CONTINO, PAUL F	

ART UNIT	PAPER NUMBER
2114	

DATE MAILED: 11/05/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/086,032	THATTE ET AL.	
	Examiner	Art Unit	
	Paul Contino	2114	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 February 2002*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-39* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-39* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *28 February 2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Claim Objections

1. Claim 10 is objected to because of the following informalities: use of the term "an" in "service an resuming" in line 3. Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

2. Claim 15 recites the limitation "the transaction grouping" and "the compensation handler" in lines 2 and 3. There is insufficient antecedent basis for this limitation in the claim when read as a dependent to claim 1.

Claim 31 recites the limitation "the customer compensation handlers". There is insufficient antecedent basis for this limitation. Examiner treats "customer" as "custom" for further consideration of claim 31.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1-39 are rejected under 35 U.S.C. 102(b) as being anticipated by Davis et al. (U.S. Patent No. 5,870,545).

As in claim 1, Davis et al. discloses an error-handling framework for business process transactions (column 5 lines 45-50), comprising:

an error-handling routine that facilitates fault correction and compensation in response to a fault associated with a business transaction (column 11 lines 35-49);

a context construct that associates a unit of work with the error-handling routine (Fig. 3; column 6 lines 35-36, 52-57, and line 62 through column 7 line 8, where a process activity of a work node is interpreted as a "context construct"; also column 12 lines 27-43); and

an execution engine (Fig. 2 #20; column 14 lines 11-12) that performs selective compensation of the unit of work upon invocation of the error-handling routine according to a set of predefined rules provided by the error-handling framework, the set of predefined rules defining propagation of error-handling in nested units of work (column 14 lines 9-60, where the combination of compensation scope and activities is interpreted as a "set of predefined rules").

As in claim 2, Davis et al. discloses the unit of work being one of a transaction and a plurality of transactions (Fig. 3; column 6 lines 35-36, 52-57, and line 62 through column 7 line 8, where each square in Fig. 3 represents a transaction or transactions).

As in claim 3, Davis et al. discloses the error-handling routine comprising an exception handler and a compensation handler (column 14 lines 11-16, where it is implied that the OpenPM engine 20 facilitates the exception handling and the compensation handling), the exception handler determines if a fault occurs and performs fault compensation if the unit of work has not completed (column 14 lines 20-22), and the exception handler calling the compensation handler to perform compensation if the unit of work has completed (column 14 lines 45-51 and column 15 lines 2-4).

As in claim 4, Davis et al. discloses a plurality of contexts associated with a plurality of units of work having at least one hierarchical relationship between units of work (Fig. 10 e.g. WN_4 in relation to WN_5), and an exception handler and a compensation handler associated with a respective context (column 14 lines 11-22, where it is implied that the OpenPM engine 20 associates the exception handling and the compensation handling with their respective "contexts"), the execution engine propagates compensation handler from outer contexts to inner contexts (column 20 lines 1-20) and exception handlers from inner contexts to outer contexts (column 14 lines 54-61 and column 21 lines 3-6).

As in claim 5, Davis et al. discloses the context construct provides support to define custom ordering of compensation handlers (column 21 lines 14-23 where the specification of the compensation activity for the process activity [in reverse order] is interpreted as "custom ordering").

As in claim 6, Davis et al. discloses the execution engine stores a snapshot of the unit of work upon completion of the unit of work, the snapshot containing data used by a compensation handler associated with the unit of work (column 24 lines 22-27, where the pre image is interpreted as a "snapshot").

As in claim 7, Davis et al. discloses the error-handling framework provides at least one of default exception handlers and default compensation handlers for contexts without custom error-handling routines (column 21 lines 41-51, where the cascading to the next save point is interpreted as "default compensation" handling; also column 5 lines 24-25).

As in claim 8, Davis et al. discloses the execution engine invoking the at least one of default exception handlers and default compensation handlers for contexts without custom error-handling routines (column 21 lines 41-51, where the cascading to the next save point is interpreted as invoking "default compensation" handling).

As in claim 9, Davis et al. discloses at least one exception handler that detects a fault and calls a compensation handler, the compensation handler calls at least one other compensation

handler (column 21 lines 14-25, where the OpenPM engine 20 acting as a compensation handler "calls" the next associated compensation handling activity when moving onto the next node to be compensated. It is interpreted that the OpenPM engine 20 embodies unique "compensation handlers" pertaining to each "context" of each fault to be compensated.).

As in claim 10, Davis et al. discloses the execution engine executing the unit of work and halting the unit of work during execution of the in-line service and resuming execution of the unit of work upon completion of the in-line call (column 9 lines 23-29, where integration on demand and activity at runtime is interpreted as an "in-line service").

As in claim 11, Davis et al. discloses default error-handling to the in-line service if custom handling is not provided, the execution engine executing one of custom error-handling and default error-handling associated with the in-line service upon detection of a fault (column 9 lines 23-29, where it is implied that the same compensation and error-handling process is applied to an activity launched at run time).

As in claim 12, Davis et al. discloses the execution engine uses functionality within the error-handling framework to determine success and failure of the unit of work when invoking the error-handling routine (column 24 lines 22-38, where success and failure is determined through comparison of the pre and post images).

As in claim 13, Davis et al. discloses a system (column 5 lines 45-50) for executing a business workflow process, comprising:

a schedule defining a business workflow process, the schedule having a business transaction grouping (column 6 lines 27-31, where a "schedule" is inherent as part of the disclosed workflow process);

a context associated with the business transaction grouping and an exception handler and at least one compensation handler associated with the context (column 14 lines 39-40, compensation scope), the exception handler defining the ordering of the at least one compensation handler (Fig. 2 #20; column 14 lines 11-16, where it is implied that the OpenPM engine 20 facilitates the exception handling and the defining the order of compensation handling; column 21 lines 14-23 where the specification of the compensation activity for the process activity [in reverse order] is interpreted as "ordering" of the "compensation handler"); and

an execution engine that executes the schedule and invokes the exception handler upon detection of a fault (column 14 lines 11-12), the exception handler performs fault correction of the business transaction grouping if the business transaction grouping has not completed (column 14 lines 20-22), and the exception handler calls the compensation handler to perform compensation of the business transaction grouping if the business transaction grouping has completed (column 14 lines 45-51 and column 15 lines 2-4).

As in claim 14, Davis et al. discloses a first compensation handler that passes a plurality of parameters to a second compensation handler (column 24 lines 39-64, where it is implied that in order to properly carry out a compensation activity, data must be "passed" between

corresponding processes [more specifically in lines 56-60]. It is interpreted that the OpenPM engine 20 embodies unique "compensation handlers" pertaining to each "context" of each fault to be compensated.).

As in claim 15, Davis et al. discloses the execution engine stores state data associated with the execution of the transaction grouping, the state data optimized is to reflect state data used by the at least one compensation handler associated with transaction grouping (column 18 lines 55-67).

As in claim 16, Davis et al. discloses a plurality of transaction groupings having at least one hierarchical relationship (Fig. 10 e.g. WN_4 in relation to WN_5), and an exception handler and a compensation handler associated with a respective transaction grouping (column 14 lines 11-16, where it is implied that the OpenPM engine 20 associates the exception handling and the compensation handling with their respective "groupings"), the execution engine propagates the execution of the compensation handlers from outer transaction groupings to inner transaction groupings (column 20 lines 1-20) and exception handlers from inner transaction groupings to outer transaction groupings (column 14 lines 54-61 and column 21 lines 3-6). It is interpreted that the compensation techniques described in columns 14, 20, and 21 may be applied to the compensation "groupings," in addition to individual work nodes, as disclosed under Lazy Compensation in column 25 lines 1-2.

As in claim 17, Davis et al. discloses at least one of the exception handler and the at least one compensation handlers being default handlers invoked by the execution engine upon detection of the absence of a custom handler (column 23 line 67 through column 24 line 3, where the compensation activity being nondeterministic is interpreted as being a "default handler"; also column 5 lines 24-25).

As in claim 18, Davis et al. discloses a method for creating a business workflow schedule, the method comprising:

defining a unit of work of a business workflow process (Fig. 10 #150);

associating a context with the unit of work (Fig. 3; column 6 lines 35-36, 52-57, and line 62 through column 7 line 8, where a process activity of a work node is interpreted as a "context"; also column 12 lines 27-43);

creating an exception handler associated with the context (Fig. 2 #20; column 14 lines 11-16, where the OpenPM engine 20 is interpreted as an "exception handler"); and

creating a compensation handler associated with the context, the compensation handler having at least one passable parameter (column 24 lines 39-64, where it is implied that in order to properly carry out a compensation activity, data must be "passed" between corresponding processes [specifically in lines 56-60]. It is interpreted that the OpenPM engine 20 embodies unique "compensation handlers" pertaining to each "context" of each fault to be compensated.).

As in claim 19, Davis et al. discloses the unit of work being one of a transaction and a plurality of transactions. (Fig. 3; column 6 lines 35-36, 52-57, and line 62 through column 7 line 8, where each square in Fig. 3 represents a transaction or transactions).

As in claim 20, Davis et al. discloses defining a plurality of transactions forming the business workflow schedule (Fig. 10 #150, 152, 154, etc.) and associating a context with respective transactions of the plurality of transactions (column 14 lines 35-43).

As in claim 21, Davis et al. discloses a plurality of transaction groupings having at least one hierarchical relationship with at least one other of the plurality of transactions (Fig. 10 e.g. WN_5 in relation to WN_2 and WN_4).

As in claim 22, Davis et al. discloses creating a plurality of compensation handlers associated with the plurality of transactions and defining the order of invocation of the plurality of transactions in response to a fault (Fig. 2 #20; column 14 lines 11-16, where it is implied that the OpenPM engine 20 facilitates the exception handling and the defining the order of compensation handling; column 21 lines 14-23 where the specification of the compensation activity for the process activity [in reverse order] is interpreted as "ordering" of the "compensation handler").

As in claim 23, Davis et al. discloses a method of executing a business workflow schedule, the method comprising:

executing a unit of work of a business workflow process defined by a business workflow schedule (column 6 lines 27-31, where a "schedule" is inherent as part of the disclosed workflow process);

storing compensation state data if the unit of work completes execution, the compensation state data providing information to be used by a compensation handler associated with the unit of work (column 18 lines 55-67);

invoking an exception handler if a fault occurs in the business workflow schedule (column 14 lines 11-12);

invoking a compensation handler if the unit of work has completed execution, the compensation handler utilizing the compensation state data to compensate for the unit of work (column 14 lines 45-51 and column 15 lines 2-4) and the exception handler compensating data associated with the unit of work if the unit of work has not completed execution (column 14 lines 20-22).

As in claim 24, Davis et al. discloses executing a plurality of units of work defined by the business workflow schedule, the plurality of units of work having at least one hierarchical relationship (Fig. 10 e.g. WN_4 in relation to WN_5).

As in claim 25, David et al. discloses the plurality of units of work having respective exception handlers and compensation handlers (column 14 lines 11-16, where it is implied that the OpenPM engine 20 associates the exception handling and the compensation handling with

their respective "units of work." It is interpreted that the OpenPM engine 20 embodies unique "compensation handlers" pertaining to each "context" of each fault to be compensated.).

As in claim 26, Davis et al. discloses associating default exception handlers to units of work without custom exception handlers and associating default compensation handlers to units of work without custom compensation handlers (column 21 lines 41-51, where the cascading to the next save point is interpreted as "default compensation" handling; also column 5 lines 24-25).

As in claim 27, Davis et al. discloses propagating compensation handlers from outer units of work to inner units of work (column 20 lines 1-20) and exception handlers from inner units of work to outer units of work in hierarchical relationships (column 14 lines 54-61 and column 21 lines 3-6).

As in claim 28, Davis et al. discloses executing a plurality of compensation handlers in an order defined in the exception handler (column 21 lines 14-23 where the specification of the compensation activity for the process activity [in reverse order] is interpreted as a "defined ordering").

As in claim 29, Davis et al. discloses halting the execution of the unit of work in response to an in-line service call, executing the service and resuming execution of the unit of work upon

completing execution of the service (column 9 lines 23-29, where integration on demand and activity at runtime is interpreted as an "in-line service").

As in claim 30, Davis et al. discloses executing a succeed component to determine if the unit of work has completed prior to invoking the compensation handler (column 24 lines 22-38, where success and failure is determined through comparison of the pre and post images).

As in claim 31, Davis et al. discloses a computer readable medium having computer executable components comprising:

a plurality of components defining a business transaction scheduling language that a user can employ to define a business transaction process (column 5 line 33 through column 6 line 24; HP OpenPM), the plurality of components including a context component that can be used to associate a unit of work with an exception handler and a compensation handler (column 8 lines 22-28, where the process activities, further described in column 14 lines 35-43, are interpreted as "context components" associated with the "handlers" of OpenPM engine 20, column 14 lines 11-14); and

a plurality of components defining an error-handling framework (column 5 lines 45-50), the error-handling framework having components for defining custom exception handlers and custom compensation handlers (column 6 lines 15-20; it is interpreted that the OpenPM engine 20 embodies unique "compensation handlers" pertaining to each "context" of each fault to be compensated), the custom compensation handlers comprising at least one passable parameter

(column 24 lines 39-64, where it is implied that in order to properly carry out a compensation activity, data must be "passed" between corresponding processes [specifically in lines 56-60]).

As in claim 32, Davis et al. discloses the error-handling framework having at least one component for defining ordering of execution of compensation handlers (column 21 lines 14-23 where the specification of the compensation activity for the process activity [in reverse order] is interpreted as a "defined ordering").

As in claim 33, Davis et al. discloses an execution engine component that utilizes the error-handling framework to provide default handlers to contexts that do not have custom handlers associated with the contexts (column 21 lines 41-51, where the cascading to the next save point is interpreted as "default compensation" handling; also column 5 lines 24-25).

As in claim 34, Davis et al. discloses an execution engine component that utilizes the error-handling framework to propagate compensation handlers from outer contexts to inner contexts (column 20 lines 1-20) and propagate exception handlers from inner contexts to outer contexts (column 14 lines 54-61 and column 21 lines 3-6).

As in claim 35, Davis et al. discloses an execution component that stores compensation data of a portion of a business process associated with a context upon completion of the execution of the portion of the business process, the compensation data provided to the

compensation handler upon invocation of a fault (column 24 lines 22-38; also column 18 lines 55-67).

As in claim 36, Davis et al. discloses a business workflow system, comprising:

means for defining a schedule of a business workflow process, the schedule having a plurality of business transaction groupings (column 6 lines 27-31, where a "schedule" is inherent as part of the disclosed workflow process);

means for defining exception handlers and compensation handlers for corresponding business transaction groupings (column 14 lines 11-16, where it is implied that the OpenPM engine 20 defines the exception handling and the compensation handling with their respective "transaction groupings." It is interpreted that the OpenPM engine 20 embodies unique "compensation handlers" pertaining to each "context" of each fault to be compensated);

means for associating the exception handlers and compensation handlers to the corresponding business transaction groupings (column 14 lines 11-16, where it is implied that the OpenPM engine 20 associates the exception handling and the compensation handling with their respective "transaction groupings"); and

means for defining the ordering of the invocation of compensation handlers in response to a fault of the business workflow process (Fig. 2 #20; column 14 lines 11-16, where it is implied that the OpenPM engine 20 facilitates the exception handling and the defining the order of compensation handling; column 21 lines 14-23 where the specification of the compensation activity for the process activity [in reverse order] is interpreted as "ordering" of the "compensation handler").

It is interpreted that the compensation techniques described in columns 14, 20, and 21 may be applied to the compensation "groupings," in addition to individual work nodes, as disclosed under the Lazy Compensation in column 25 lines 1-2.

As in claim 37, Davis et al. discloses means for propagating compensation handlers and exception handlers for business transaction groupings having hierarchical relationships (Fig. 10 e.g. WN_5 in relation to WN_2 and WN_4),

As in claim 38, Davis et al. discloses providing default handlers in the absence of custom handlers (column 23 line 67 through column 24 line 3, where the compensation activity being nondeterministic is interpreted as being a "default handler"; also column 5 lines 24-25).

As in claim 39, Davis et al. discloses means for storing compensation data upon completion of execution of a business transaction grouping (column 18 lines 55-67).

Conclusion

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Paul Contino whose telephone number is (571) 272-3657. The examiner can normally be reached on Monday-Friday 7:30 am - 5:00 pm, first Fridays off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Robert Beausoliel can be reached on (571) 272-3645. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3657.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

PFC
October 26, 2004

SCOTT BADERMAN
PRIMARY EXAMINER